Date: Tue, 16 Jun 1992 17:05:23 +0200
From: Steven.Pemberton@cwi.nl
Message-Id: <9206161505.AA06927.steven@sijs.cwi.nl>
To: berliner@Sun.COM
Subject: cvs

INTRODUCTION TO USING CVS

CVS is a system that lets groups of people work simultaneously on groups of files (for instance program sources).

It works by holding a central 'repository' of the most recent version of the files. You may at any time create a personal copy of these files; if at a later date newer versions of the files are put in the repository, you can 'update' your copy.

You may edit your copy of the files freely. If new versions of the files have been put in the repository in the meantime, doing an update merges the changes in the central copy into your copy.
(It can be that when you do an update, the changes in the central copy clash with changes you have made in your own copy. In this case cvs warns you, and you have to resolve the clash in your copy.)

When you are satisfied with the changes you have made in your copy of the files, you can 'commit' them into the central repository.
(When you do a commit, if you haven't updated to the most recent version of the files, cvs tells you this; then you have to first update, resolve any possible clashes, and then redo the commit.)

USING CVS

Suppose that a number of repositories have been stored in /usr/src/cvs. Whenever you use cvs, the environment variable CVSROOT must be set to this (for some reason):

```
CVSROOT=/usr/src/cvs
export CVSROOT
```

TO CREATE A PERSONAL COPY OF A REPOSITORY

Suppose you want a copy of the files in repository 'views' to be created in your directory src. Go to the place where you want your copy of the directory, and do a 'checkout' of the directory you want:

```
cd $HOME/src
cvs checkout views
```

This creates a directory called (in this case) 'views' in the src directory, containing a copy of the files, which you may now work on to your heart's content.

TO UPDATE YOUR COPY

Use the command 'cvs update'.

This will update your copy with any changes from the central repository, telling you which files have been updated (their names are displayed with a U before them), and which have been modified by you and not yet committed (preceded by an M). You will be warned of any files that contain clashes, the clashes will be marked in the file surrounded by lines of the form <<<< and >>>>.

TO COMMIT YOUR CHANGES

Use the command 'cvs commit'.

You will be put in an editor to make a message that describes the changes that you have made (for future reference). Your changes will then be added to the central copy.

ADDING AND REMOVING FILES

It can be that the changes you want to make involve a completely new file, or removing an existing one. The commands to use here are:

```
cvs add <filename>
cvs remove <filename>
```

You still have to do a commit after these commands. You may make any number of new files in your copy of the repository, but they will not be committed to the central copy unless you do a 'cvs add'.

OTHER USEFUL COMMANDS AND HINTS

To see the commit messages for files, and who made them, use:

```
cvs log [filenames]
```

To see the differences between your version and the central version:

```
cvs diff [filenames]
```

To give a file a new name, rename it and do an add and a remove.

To lose your changes and go back to the version from the repository, delete the file and do an update.

After an update where there have been clashes, your original version of the file is saved as .#file.version.

All the cvs commands mentioned accept a flag '-n', that doesn't do the action, but lets you see what would happen. For instance, you can use 'cvs -n update' to see which files would be updated.

MORE INFORMATION

This is necessarily a very brief introduction. See the manual page (man cvs) for full details.